Research on Web Service Dynamic Selection with Multi-constraint

Heng-wei ZHANG, Chen FANG and Tao LI
Zhengzhou Institute of Information Science and Technology
Zhengzhou, China
*Corresponding author

Keywords: Service composition, Multiple constraints, Decomposition of global QoS.

Abstract: With the massive growth of Web services in cloud computing environment, users require as much short time as possible to get high-quality composite service that meets multiple constraints. Therefore, this paper proposes a multi-constraint service selection method based on decomposition of global QoS. The method uses cultural genetic algorithm to decompose the global QoS constraints, and then combines with the dependency relationship and compatible relationship among candidate services to filter out the unsatisfied services. Finally the most appropriate composite service is obtained by local selection. Experimental results prove the effectiveness of this method.

Introduction

With the rapid growth of Web services in the Internet, more and more services have similar functional properties. While selecting services, users put forward requirements for the quality of composite service (QoS). However, QoS is uncertain in dynamic Internet. So methods which can quickly select appropriate services are of great importance to ensure the quality and performance of composite services.

Reference [1] decomposed global QoS constraint by mixed integer programming, and got the most appropriate composite service by local selection. Reference [2] proposed an adaptive adjustment method based on fuzzy logic to find the optimal number of decomposition satisfying users’ preferences, and improved the accuracy of service selection. However, the above service selection method only considered global QoS constraints, but ignored functional constraints that may exist between the candidate services. To solve this problem, this paper proposes a dynamic service selection method with multi-constraints based on global QoS decomposition (referred to as DSS-QD).

Question Modeling

Service composition based on the global QoS constraints and functional constraints is an optimization problem, also an NP-hard problem. This paper establishes a single objective optimization model with multiple constraints as follows:

\[
U(CS) = \max \sum_{k=1}^{r} w_k \times q_k(CS)
\]

\[
\begin{align*}
\sum_{j=1}^{n} x_{ij} &= 1, \quad x_{ij} \geq 0 \quad (1 \leq j \leq m) \\
\text{s.t.} \quad x_{ab} - x_{cd} &\leq 0 \quad (a, b, c, d \ satisfies(s_{ad}, s_{ub}, \oplus)) \\
\quad x_{ef} + x_{gh} &\leq 1 \quad (e, f, g, h \ satisfies(s_{ef}, s_{gh}, \otimes)) \\
q_k(CS) &\leq C_k \quad (1 \leq k \leq r)
\end{align*}
\]

The description of parameters in Formula(1) is as follows:

(1) \( w_k (1 \leq k \leq r) \) is the weight of QoS attribute \( k \), and \( \sum_{k=1}^{r} w_k = 1 \). CS indicates a composite service.
composed of \( m \) basic service classes \( S_j \). \( x_{ij} \) indicates the state of \( i \)-th candidate service of basic service class \( S_j \). \( x_{ij} = 1 \) indicates the service is selected, and \( x_{ij} = 0 \) indicates the service isn’t selected. \( C = \{ C_1, C_2, \ldots, C_n \} \) indicates the global QoS constraints. \( q_k(C) \leq C_k \) indicates the \( k \)-th QoS value of composite service has to meet the corresponding global constraints.

\( s_{ab} \) indicates the \( b \)-th candidate service of basic service class \( S_a \). Dependency relationship \( t_i = (s_{cd} , s_{ab}, \Theta) \) indicates service \( s_{ab} \) depends on service \( s_{cd} \), \( x_{cd} - x_{cd} \leq 0 \) ensures that only when service \( s_{cd} \) is selected, service \( s_{ab} \) is perhaps to be selected. Conflicting relationship \( d_i = (s_{ef} , s_{gh}, \Theta) \) indicates \( s_{ef} \) is incompatible with \( s_{gh} \), \( x_{ef} + x_{gh} \leq 1 \) ensures that \( s_{ef} \) and \( s_{gh} \) can’t be simultaneously selected.

**Service Selection Method**

In this paper, we firstly decompose the global QoS constraint into the local QoS constraint for each service class, then combines with the functional constraints to filter out the unsatisfied services, finally get the optimal composite service by local selection. The specific steps are as follows.

**Step 1:** According to the transitivity of dependency, we establish dependency set \( t(s_j) \) and conflicting set \( d(s_j) \) for each candidate service \( s_j \). The transfer rules of dependency are as follows:

\[
\begin{align*}
(s_j, s_{ij}, \Theta) &\Rightarrow s_{ij} \in t(s_j) & (s_j, s_{ij}, \Theta) &\Rightarrow s_{ij} \in d(s_j) \\
(s_j, s_{ij}, \Theta) \land (s_i, s_{ij}, \Theta) &\Rightarrow s_{ij} \in t(s_i) & (s_j, s_{ij}, \Theta) \land (s_i, s_{ij}, \Theta) &\Rightarrow s_{ij} \in d(s_i) \land s_{ij} \in d(s_i) \\
\end{align*}
\]

(2)

The 1st formula indicates that if \( s_{ij} \) relies on \( s_i \), then \( s_{ij} \) belongs to the dependency set \( t(s_i) \) of \( s_i \); The 2st formula indicates that if \( s_i \) is incompatible with \( s_{ij} \), then both are included into each other’s conflicting set; The 3st formula indicates that if \( s_{ij} \) relies on \( s_i \), \( s_{ij} \) relies on \( s_i \), then \( s_{ij} \) belongs to the dependency set \( t(s_i) \) of \( s_i \); The 4st formula indicates that if \( s_{ij} \) relies on \( s_i \), \( s_i \) is incompatible with \( s_{ij} \), then \( s_{ij} \) is also incompatible with \( s_i \).

**Step 2:** Using an improved culture genetic algorithm (CGA) to decompose the global QoS constraint into local QoS constraint corresponding to each service class, then filter out those candidate services which don’t meet the local QoS constraint. More details about the algorithm are described later.

**Step 3:** Check all the filtered candidate services. If a candidate service is included in the filtered service’s dependency set, then filter out the candidate service too, and update the remaining candidate services’ dependency set and conflicting set. This is the second filtering stage.

**Step 4:** Sort the remaining services according to their quality, and select the candidate service that has the largest local fitness, and finally form the most appropriate composite service.

\[
Q(s_j) = \sum_{i=1}^{r} w_i \times \frac{q_k(S_j) - q_k(S_j)}{q_{max}(S_j) - q_{max}(S_j)}
\]

(3)

**Global QoS Decomposition**

**The Framework of CGA**

Genetic algorithm (GA) searches for solutions by simulating natural evolution operations such as selection, crossover, mutation, etc. Culture algorithm (CA) is composed of population space and belief space, respectively simulating the evolution of culture from microscopic and macroscopic views. This paper proposes a culture genetic algorithm (CGA). The basic idea is embedding the evolution of GA into the population space of CA. Its framework is shown as Figure 2.
Design of fitness function and encoding

$C_k$ is the $k$-th QoS constraint, $q_r(s_m)$ denotes $r$-th QoS value of $n$-th candidate service in service class $S_j$. $L_{jk}^i$ indicates $d$-th quality level of $r$-th QoS in service class $S_j$, and it satisfies $q_{k_{max}}^i(S_j) \leq L_{jk}^i \leq L_{jk}^{i+1} \leq \cdots \leq L_{jk}^{i+d} \leq q_{k_{max}}^{i+d}(S_j)$, where $d$ is the number of quality levels. The decomposition of quality levels is shown as Figure 3:

**Definition 1:** quality level weight $b_{jk}^i$. It indicates the probability of candidate services being selected when selecting quality level $L_{jk}^i$ for $k$-th QoS in service class $S_j$, which is computed as Formua(4).

$$b_{jk}^i = \frac{n(L_{jk}^i) \cdot Q_{max}(L_{jk}^i)}{n(S_j) \cdot Q_{max}(S_j)} , 1 \leq i \leq d$$ \hspace{1cm} (4)

Where $n(L_{jk}^i)$ is the number of services contained in the quality level $L_{jk}^i$. $n(S_j)$ is the total number of services in $S_j$. $Q_{max}(L_{jk}^i)$ is the maximum utility function value of services contained in $L_{jk}^i$. $Q_{max}(S_j)$ is the maximum utility function value of services in service class $S_j$. In order to find optimal quality level combination for each service class, we build a single objective optimization model as follows:

$$\max f = \sum_{j=1}^{m} \sum_{k=1}^{r} \sum_{i=1}^{d} \ln(b_{jk}^i) \cdot x_{jk}^i$$

$$\forall k: \sum_{j=1}^{m} \sum_{i=1}^{d} L_{jk}^i x_{jk}^i \leq C_k, 1 \leq k \leq r$$

$$\forall j, k: \sum_{i=1}^{d} x_{jk}^i = 1, 1 \leq j \leq m, 1 \leq k \leq r$$ \hspace{1cm} (5)

Where $m$ is the total number of service classes, $r$ is the number of QoS attributes, $d$ is the number of quality levels. The result of global QoS decomposition is getting $m$ optimal quality level combinations. Based on this, this paper designs a $m$-dimensional chromosome model to represent the $m$ quality level combinations, which is shown as Figure 4. Our goal is to get the optimal ones.
Global QoS decomposition algorithm based on CGA.

Global QoS Decomposition Algorithm Based on CGA

**Input:** The number of service classes \( m \), QoS values of services, the number of quality levels \( d \), global QoS constraint \( C_k \) \((1 \leq k \leq r)\), the scale of population space \( N \), crossover probability \( p_1 \), mutation probability \( p_2 \), the number of excellent solutions passed to the belief space \( q \), the accumulated updating times in belief space \( \eta \), maximum iteration times \( M \).

**Output:** the optimal quality level combination for each service class.

**Step 1: Initialization phase**
1. Randomly generate \( N \) initial valid individuals in the population space; evaluate all the individuals by fitness function.

**Step 2: Evolution in the population space**
2. Individuals in the population space carry out selection, crossover, mutation operations by \( PEvolve() \) function.

**Step 3: Evolution in the belief space**
3. Select \( q \) excellent individuals from the population space, and pass them to the belief space by \( accept() \) function.
4. Replace the poor individuals in the belief space with better individuals passed from the population space by \( update() \) function.

**Step 4: Guide the evolution of the population space**
5. When the individuals in the belief space has been updated for \( \eta \) times, replace the \( q \) poor individuals in the population space with \( q \) excellent individuals in the belief space by \( Influence() \) function.

**Step 5: Whether stop to evolve**
6. If (Iteration times=\( M \)) Then
   Output the optimal solutions in the belief space and stop computing
   End if
7. Else
   Iteration times+1, go to Step2.
   End if

**Experimental Studies**

The test compares the proposed method with three other service selection methods with QoS constraint, respectively GS in [3], HGA in [4], LDPSO in [5]. This paper’s method is DSS-QD. QWS datasets are used in the experiments. Global QoS constraints are: (1) response time<2s (2) availability>0.4 (3) reliability>0.4 (4) cost<100. \( p_1 = 0.85 \), \( p_2 = 0.05 \), \( q = 20 \), \( \eta = 5 \), \( M=200 \), \( d = 10 \).

**The Performance under Different Scale of Candidate Services**

Figure 5 and 6 shows the execution time and composite service’s fitness of four methods under different scale of candidate services. The optimal quality level returned by CGA can help select better candidate services, so DSS-QD performs better than the other three methods.
B. The performance under different scale of service classes

Figure 5. The change of execution time.

Figure 6. The change of fitness.

Figure 7. The change of execution time.

Figure 8. The change of fitness.

Figure 7 and 8 shows the execution time and composite service’s fitness of four methods under different scale of service classes. By combining the functional constraints with the QoS constraints to filter out the unsatisfied services, the final composite service is of better quality.

Conclusion

This paper proposes a dynamic service selection method with multi-constraint based on global QoS decomposition. We design a culture genetic algorithm (CGA) to decompose the global QoS constraint to local QoS constraint. Then by combining with the dependency and conflicting relationships between services, it filters out those unsatisfied services and calculate the fitness of remaining services. The optimal composite service is finally obtained by local selection.

References


